Exam, Week 3

Name:

Name of student to the left:

Name of student to the right:

Directions

- **This exam contains 18 questions and will last 90 minutes.**
- Two questions are long answer, while the other 16 questions have you select from multiple choices or have you write a short answer.
- Use your time wisely. If you are having too much trouble on a question, skip it and return to it later. **Avoid getting stuck.**
- In the answer options, the ↵ symbol indicates a new line. The ↵ symbol will only be used to separate lines of output and will not appear at the end of the final line.
- For questions with *circular bubbles*, you should select *exactly one* choice.
  - ○ You must choose either this option
  - ○ Or this one, but not both!
- For questions with *square checkboxes*, you may select *multiple* choices.
  - □ You could select this choice.
  - □ You could select this one too!

Staff use only.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Q12	Q13	Q14	Q15	Q16	Q17	Q18				Total

1. Consider the following lines of code.

```
n = 0
while n < 5:
    n += 1
    print(n)
```

What will be printed?

- ○ 1 ↵ 2 ↵ 3 ↵ 4 ↵ 5
- ○ 1 ↵ 2 ↵ 3 ↵ 4
- ○ 2 ↵ 3 ↵ 4
- ○ 2 ↵ 3 ↵ 4 ↵ 5

2. Fill in the blanks so that the desired strings are printed.

```
def join_digits(________, ________):
        x = str(________)
        y = str(________)
        return x + y

print(join_digits(________, ________))  # Should print "2024"
print(join_digits(________, ________))  # Should print "99"
```

3. What is printed when the following code is run?

```
def happy_face():
    print(":)")
    return ":)"

print(happy_face())
happy_face()
```

○ :)

○ :) ↵ :)

○ :) ↵ :) ↵ :)

○ :) ↵ :) ↵ :) ↵ :)

4. For which value(s) of `a` and `b` will the following code print `Timnit`?

```
if a:
    print('Boaz')
elif b:
    print('Timnit')
else:
    print('Jelani')
```

Fill in the boxes next to all answers that print `Timnit`.

□ `a = True`, `b = True`

□ `a = True`, `b = False`

□ `a = False`, `b = True`

□ `a = False`, `b = False`

5. What is printed after running the following lines of code? Write your answers on the dashed lines.

```
x = 5
x = x // 2
print(x)  # Printed: __________
x = x * 2
print(x)  # Printed: __________
x = x % 3
print(x)  # Printed: __________
x = x / 2
print(x)  # Printed: __________
```

6. Consider the following lines of code.

```
x = 8
if x > 6 and x < 5:
    print("Great")
else:
    print("Good")
if x > 6 or x < 5:
    print("TA")
else:
    print("Student")
```

What will be printed?

○ `Great ↵ TA`

○ `Great ↵ Student`

○ `Good ↵ TA`

○ `Good ↵ Student`

7. Consider the following code.

```
lst =  [1, 2, 3]
lst *= 2
lst += ["salt" + "fish"]
print(lst)
```

What is printed on the last line below?

- ○ `[2, 4, 6, "saltfish"]`
- ○ `[2, 4, 6, "salt", "fish"]`
- ○ `[1, 2, 3, 1, 2, 3, "salt", "fish"]`
- ○ `[1, 2, 3, 1, 2, 3, "saltfish"]`

8. What is printed after running the following lines of code? Write your answers on the dashed line.

```
n = 14
while n >= 0:
    n -= 3
print(n) # Printed: __________
```

9. What is printed after running the following lines of code? Write your answers on the dashed lines.

```
a = [1, 10, 100]
b = [9, 99, 999]
a += b[:2]
print(a)       # Printed: _____________________
print(b)       # Printed: _____________________
b += a[2:]
print(a)       # Printed: _____________________
print(b)       # Printed: _____________________
print(b[a[0]]) # Printed: _____________________
```

10. The variables `a`, `b`, and `c` are defined as:

```
a = 1
b = "2"
c = 3
```

Fill in the blanks. If there would be an error, write `error`.

- The type of `a` is __________. The type of `b` is __________.
- Running `print(a + b)` displays __________.
- Running `print(a * b)` displays __________.
- Running `print(a + c)` displays __________.
- Running `print(a * c)` displays __________.

11. Fill in the blanks below so that `find_anita(lst)` returns the index of the string `"Anita"` in `lst`. If there is no `"Anita"` in the list, return `None`.

```
def find_anita(lst):
    for ________________________:
        if ________________________:
            return ________________________
    return None

print(find_anita(["James", "Anita", "Tarun"]))  # Prints 1
print(find_anita(["Xavier", "Ecy"]))            # Prints None
```

12. Fill in the blanks to write a function `sum_positive` that returns the sum of the positive integers in a list.

```
    def sum_positive(lst):
        result = _____________
        for _____________ in _____________:
            if _____________:
                result += _____________
        return result
print(sum_positive([1, -10, 5]))  # Should print 6
```

13. What will be printed after running the following code? Write your answers on the dashed lines.

```
def excited(word):
    print(word + "!")
    return word

def confused(word):
    print(word + "?")

def bored(word):
    return word + "..."

confused(bored("what"))       # Printed: _____________________________
print(excited(bored("huh"))) # Printed: _____________________________
confused(excited("yay"))      # Printed: _____________________________
```

14. Consider the following lines of code.

```
def cat(x):
    print("meow")
    if x <= 0:
        return
    dog(x-1)

def dog(x):
    print("woof")
    cat(x-1)

cat(2)
```

What will be printed when this code is run?

○ `meow ↵ woof`

○ `meow ↵ woof ↵ meow`

○ `meow ↵ woof ↵ meow ↵ woof`

○ `meow ↵ woof ↵ meow ↵ woof ↵ meow ↵ woof ↵ ...` repeatedly, until an error occurs.

15. Consider the following lines of code.

```
def mystery(lst):
    if len(lst) == 0:
        return 0
    if lst[0] < 10:
        return mystery(lst[1:]) - 1
    else:
        return mystery(lst[1:])

print(mystery([1, 6, 19, 24]))
```

What will be printed when this code is run?

○ 2

○ [0, 5]

○ -2

○ [0, 5, 18, 23]

16. What is printed after running the following lines of code? Write your answers on the dashed lines.

```
def tricky(lst, x):
    for i in range(len(lst)):
        if (i * lst[i]) % x != 0:
            print(lst[i])
tricky([1, 2, 3, 4], 2)    # Printed ______________________________
tricky([8, 4, 2, 1], 3)    # Printed ______________________________
```

17. A list of integers is *consecutive* if it is of the form `[x, x+1, x+2, ...]` for some `x`. In other words, if each integer in the list is larger than its previous element by 1.

Write a function `is_con` that is given a list of integers `lst` and returns `True` if it is consecutive, and returns `False` otherwise.

```
For example:
Argument: lst = [17, 18, 19, 20, 21]
Returns: True
Argument: lst = [5, 6, 7, 10]
Returns: False
```

```python
def is_con(lst):
    """
    Args: lst (list of int)
    Returns (boolean): True if lst is consecutive, False otherwise.
    """
```

18. Write a function count_char which takes a list of strings lst and a character char, and returns the total number of times char appears in all strings in lst.

```
For example:
Arguments: lst = ["zaria", "anita"], char = "a"
Returns: 4
Arguments: lst = ["jamcoders", "2024", "rocks"], char = "j"
Returns: 1
```

```python
def count_char(lst, char):
    """
    Args: lst (list of str), char (str)
    Returns (int): Total number of times char occurs in
                   each string in lst.
    """
```

19. Bonus Question: First Infected Sample

You are the head scientist in a research lab, leading a team collecting data on various DNA samples. Unfortunately, your latest sample has failed the purity test. Since each collected sample is from the same pool, all samples after the impure sample are also unusable. Additionally, there are hundreds of thousands of samples, so checking each individually is an impossible task.

Suppose there are n samples in a list `sample_list [1, 2, ..., n]`, and you must find the first impure sample that is causing the rest to be impure.

You are given an arbitrary function `isImpure(sample)` which takes an integer `sample` and returns a `Boolean (True or False)` of whether the sample is impure.

Write a function to find the first impure sample.

```
Example:
Input: sample_list = [1, 2, ..., 457000], bad = 567
Output: 567
isImpure(2500) -> True
isImpure(230) -> False
IsImpure(567) -> True
```

Bonus bonus question: What is the time complexity of the optimal solution?